

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam kerja magang yang telah dilakukan, kedudukan yang diterima adalah sebagai *Junior Hybrid Programmer* dalam mengembangkan *plugin manage stories*. Selama proses pengembangan *plugin*, dibimbing oleh Bapak Jason Japutra selaku kepala divisi Hybrid. Koordinasi selama proses kerja magang diberikan arahan oleh Bapak Jason Japutra serta dilakukan evaluasi setiap seminggu sekali.

3.2 Tugas yang dilakukan

Tugas yang dilakukan adalah mengembangkan *plugin manage stories* berbasis web dari klien PT Mirai Internasional Indonesia dengan menggunakan HTML, CSS, PHP, JQuery, dan AJAX. Dalam proses pengembangannya, Pengelolaan data pada sistem yang dikembangkan menggunakan *database MySQL*. Sebelum melakukan proses pengerjaan proyek, pembimbing lapangan memberikan sedikit pelatihan terkait dengan proyek yang akan dikerjakan nantinya. Setelah diberikan pelatihan, dibuatlah perencanaan kerja dengan membuat sebuah tabel *timeline* untuk pengembangan *plugin* berbasis web PT Internasional Indonesia ini. *Timeline* pengembangan *plugin manage stories* PT Internasional Indonesia dapat dilihat pada Tabel 3.1

Tabel 3.1 Timeline Kerja Magang

No	Kegiatan	Minggu							
		1	2	3	4	5	6	7	8
1	Studi Literatur								
2	Analisis <i>Requirement</i>								
3	Analisis Sistem								
4	Implementasi								
5	Testing								
6	Dokumentasi dan Penulisan Laporan Magang								

Dalam prosesnya dilakukan studi literatur sebelum dan saat mengerjakannya, karena dibutuhkan referensi dalam pengembangan *plugin* tersebut serta sebagai proses pembelajaran hal-hal baru dan menambah wawasan yang belum diketahui sebelumnya. Kemudian, dilakukan analisis terkait *requirement* yang dibutuhkan dalam proses pengembangan proyek sehingga ke depannya bisa lebih efektif dalam pengerjaan proyek tersebut. Selanjutnya, dilakukan analisis terhadap sistem yang akan dikembangkan untuk mempermudah proses implementasi. Setelah dilakukan analisis sistem, maka dilanjutkan ke tahap implementasi. Selama proses pengembangan *plugin* tersebut, terdapat beberapa perangkat keras dan perangkat lunak yang digunakan. Berikut ini adalah perangkat-perangkat keras yang digunakan selama proses pengembangan dan implementasi. Spesifikasi *Hardware* yang digunakan adalah laptop Asus A455L dengan RAM 4GB, Intel (R) Core (TM) i5-4210U CPU @1.70GHz (4 CPUs). Software yang digunakan adalah sebagai berikut.

1. XAMPP 5.6.8
2. MYSQL
3. JetBrains PhpStorm 8.0.1
4. Wordpress
5. Google Chrome

Setelah proses pengembangan *plugin* selesai, dilakukan proses pengujian untuk mengetahui apakah *plugin* yang dikembangkan sudah berjalan dengan baik atau masih terdapat *error*. Dilakukan tahap pengujian dalam pengembangan *plugin* tersebut. Tahap pertama melakukan pengujian terhadap kode-kode program yang telah dibuat apakah dapat berjalan dengan baik dan sesuai dengan keinginan pembimbing lapangan atau tidak. Jika ada kekurangan, akan dilakukan revisi terhadap sistem tersebut.

Setelah lewat tahap pengujian dan sesuai dengan keinginan pembimbing lapangan, maka pembimbing lapangan akan meminta *file-file* yang telah dikembangkan untuk dilakukan evaluasi lebih lanjut sampai akhirnya sistem siap untuk digunakan.

Tahap terakhir yang dilakukan adalah membuat dokumentasi secara keseluruhan selama proses kerja magang dilakukan, serta penulisan laporan kerja magang selama proses pengembangan *plugin manage stories*.

3.3 Uraian Pelaksanaan Kerja Magang

Pada bagian ini dijelaskan tentang proses pelaksanaan kerja magang yang berkaitan dengan pembuatan proyek, perancangan sistem serta implementasi yang telah dilakukan selama proses kerja magang.

3.3.1 Proses Pelaksanaan

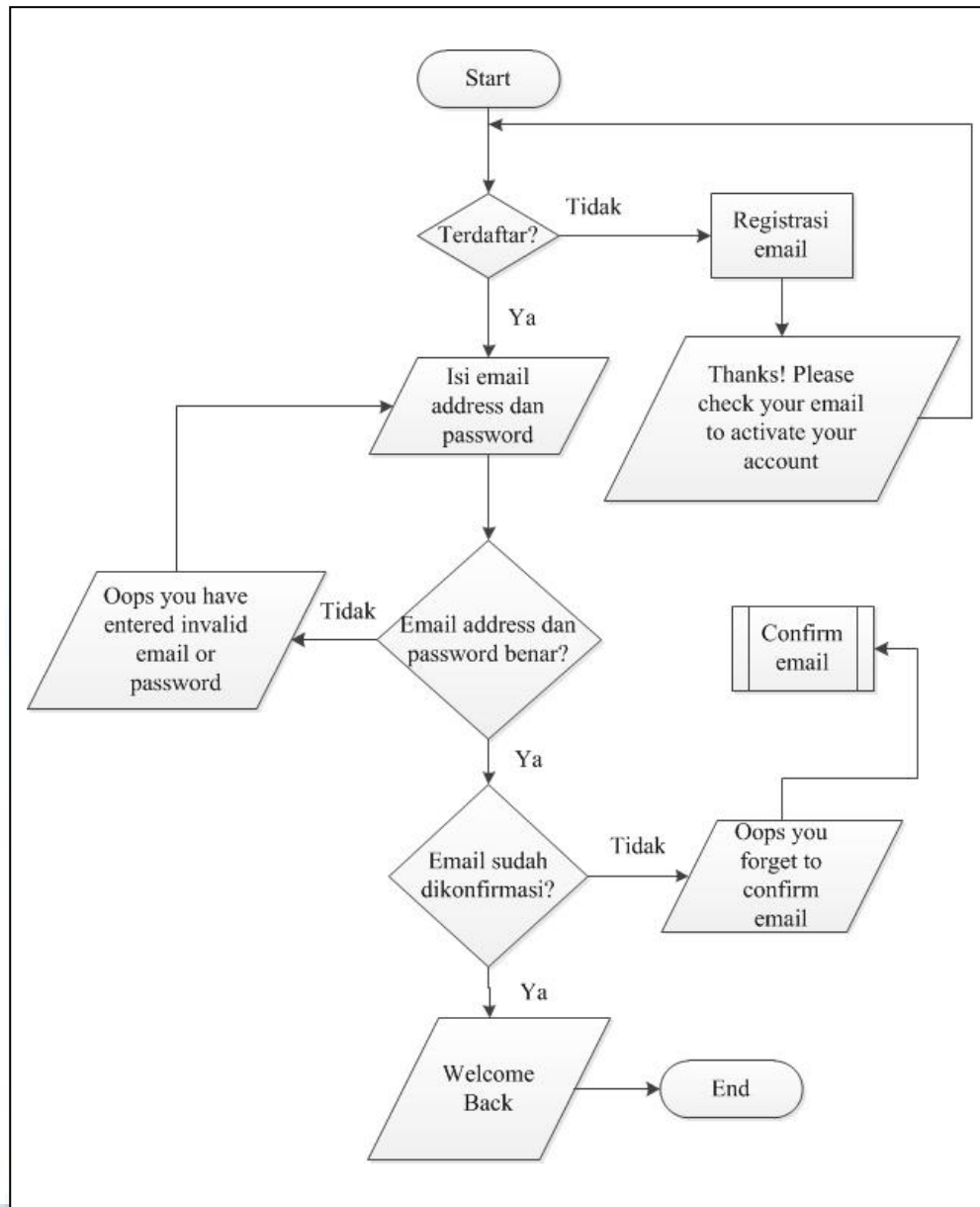
Dalam proses pelaksanaan kerja magang terdapat 2 tugas lain dari tugas utama kerja magang yaitu pembuatan *flowchart* untuk aplikasi karpul dan dokumentasi QA(Quality Assurance) aplikasi Karpul. Tugas utama yang dikerjakan adalah mengembangkan *plugin manage stories* dengan Wordpress menggunakan php. Pekerjaan yang dilakukan adalah sebagai berikut.

1. Pembuatan *Flowchart* untuk aplikasi Karpul

- 1.1 Flowchart Register dan Sign in

Sebelum masuk ke sistem Karpul diperlukan *register* untuk membuat akun. Jika sudah memiliki akun maka diperlukan *sign in* untuk masuk ke dalam aplikasi Karpul.

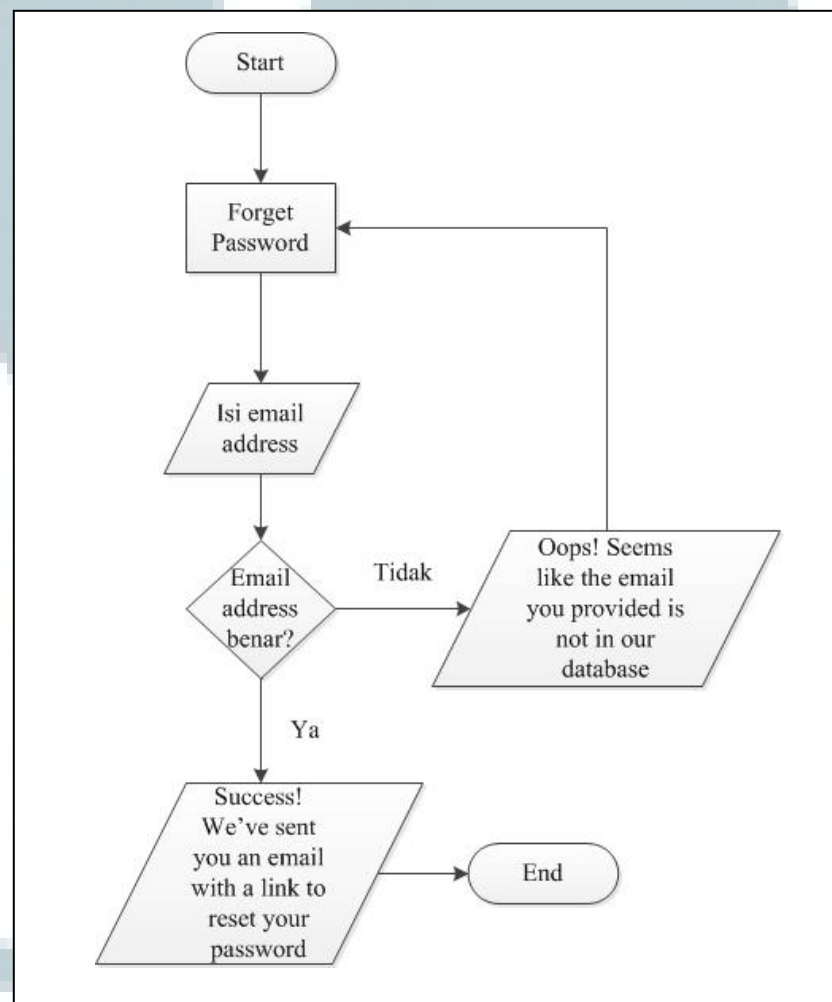
UMN



Gambar 3.1 *Flowchart Register and Sign in*

1.2 Flowchart Forget Password

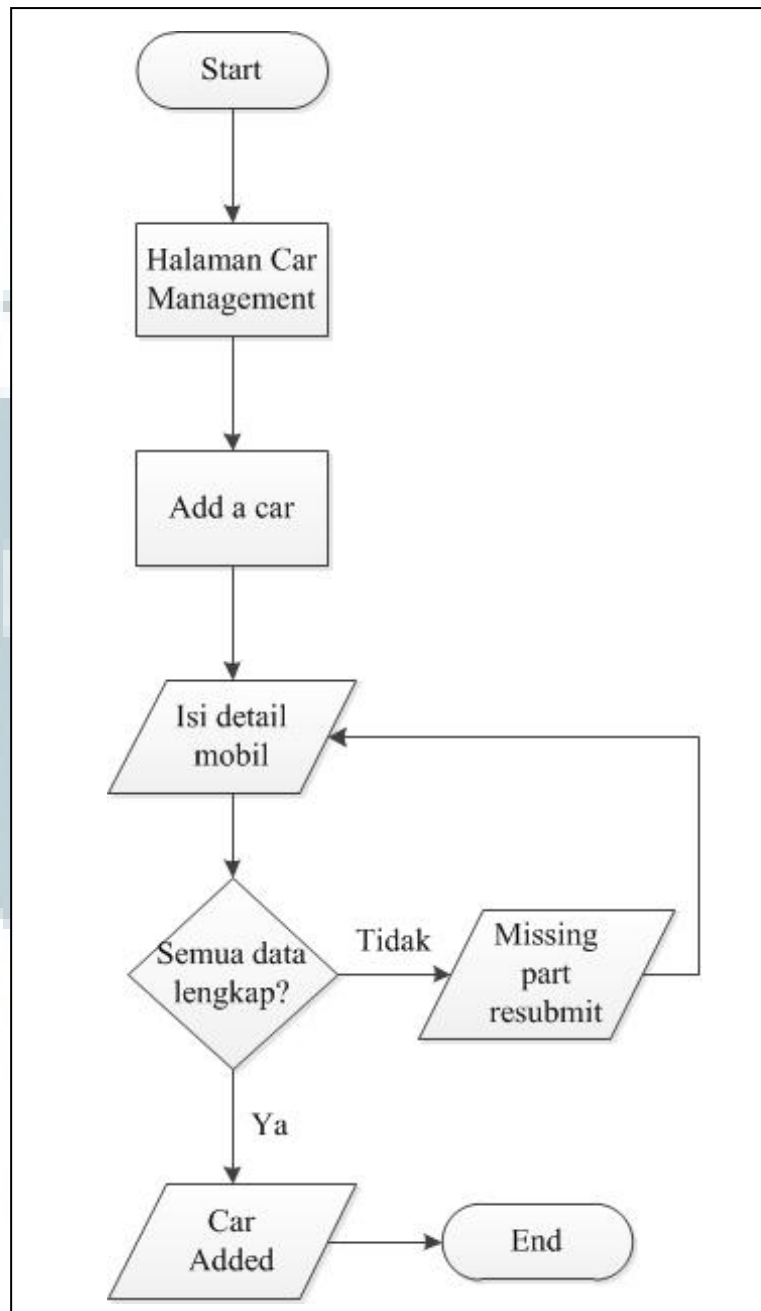
Flowchart pada gambar 3.2 menjelaskan mengenai proses *Forget Password* dimana proses ini dimulai dengan user menekan tombol *Forget Password* dan mengisi *email address*. Setelah *user* mengisi email address maka sistem akan melakukan verifikasi terhadap email apakah email terdaftar atau tidak. Jika terdaftar maka akan muncul kotak *popup* “*Success! We’ve sent you an email with a link to reset your password*”, Jika tidak terdaftar maka akan muncul *Pop up box* “*Oops! Seems like the email you provided not in our database*”.



Gambar 3.2 *Flowchart Forget Password*

1.3 Flowchart Add Car –Driver

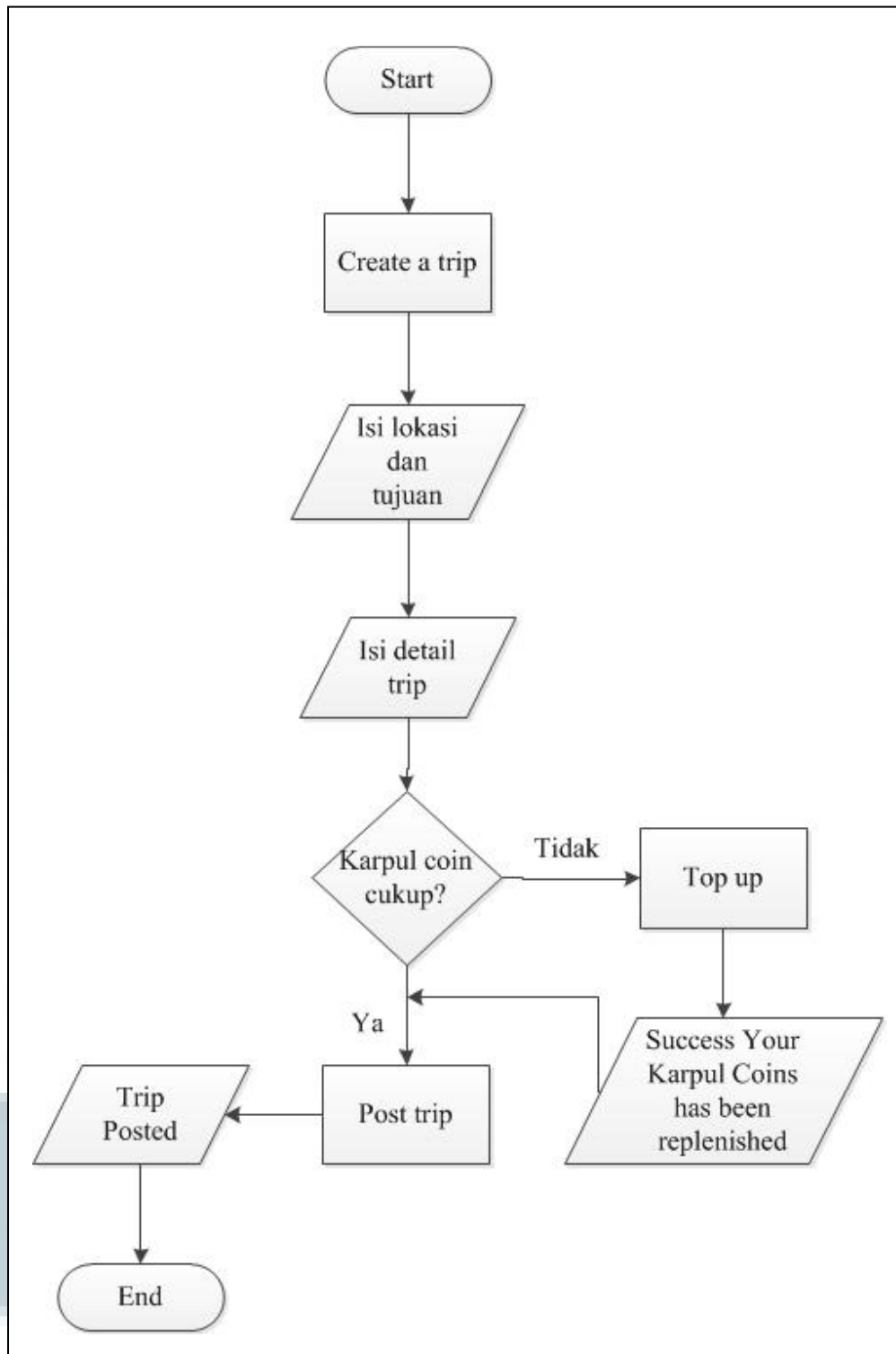
Flowchart pada Gambar 3.3 menjelaskan mengenai proses *Add Car* dimana proses dimulai dengan *user* menekan tombol *Car Management* untuk masuk halaman *Car Management*, kemudian *user* menekan tombol *Add a car* untuk masuk halaman *Add Car*, lalu *user* memasukkan data-data tentang mobil dan menekan tombol *Save*, Setelah proses *Add Car* selesai sistem akan melakukan pengecekan terhadap data yang dimasukkan oleh *user*. Jika data lengkap maka akan muncul kotakpopup “*Car Added*” dan data mobil otomatis masuk ke *Car Management Page*, Jika data tidak lengkap maka akan muncul pesan *error*.



Gambar 3.3 Flowchart Add Car

1.4 Flowchart Post Trip –Driver

Flowchart pada Gambar 3.4 menjelaskan proses *Post Trip* dimana proses dimulai dari *user* menekan tombol create a trip, setelah menekan tombol create a trip maka *user* diminta memasukkan lokasi dan tujuan, lalu pada halaman berikutnya *user* diminta memasukkan detail perjalanan, waktu, dan jenis mobil. Setelah *user* selesai memasukkan data maka *user* diminta menekan tombol next untuk lanjut ke halaman berikutnya. Pada halaman berikutnya akan ada pengecekan Karpul Coin milik *user*. Jika Karpul Coin cukup maka user bisa menekan tombol Post Trip dan akan muncul kotak *pop up* “*Trip Posted*”, Jika Karpul Coin tidak cukup maka user diminta menekan tombol Top Up Setelah proses Top Up selesai maka akan muncul *Pop up box* “*Success! Your Karpul Coins has been replenished.*”

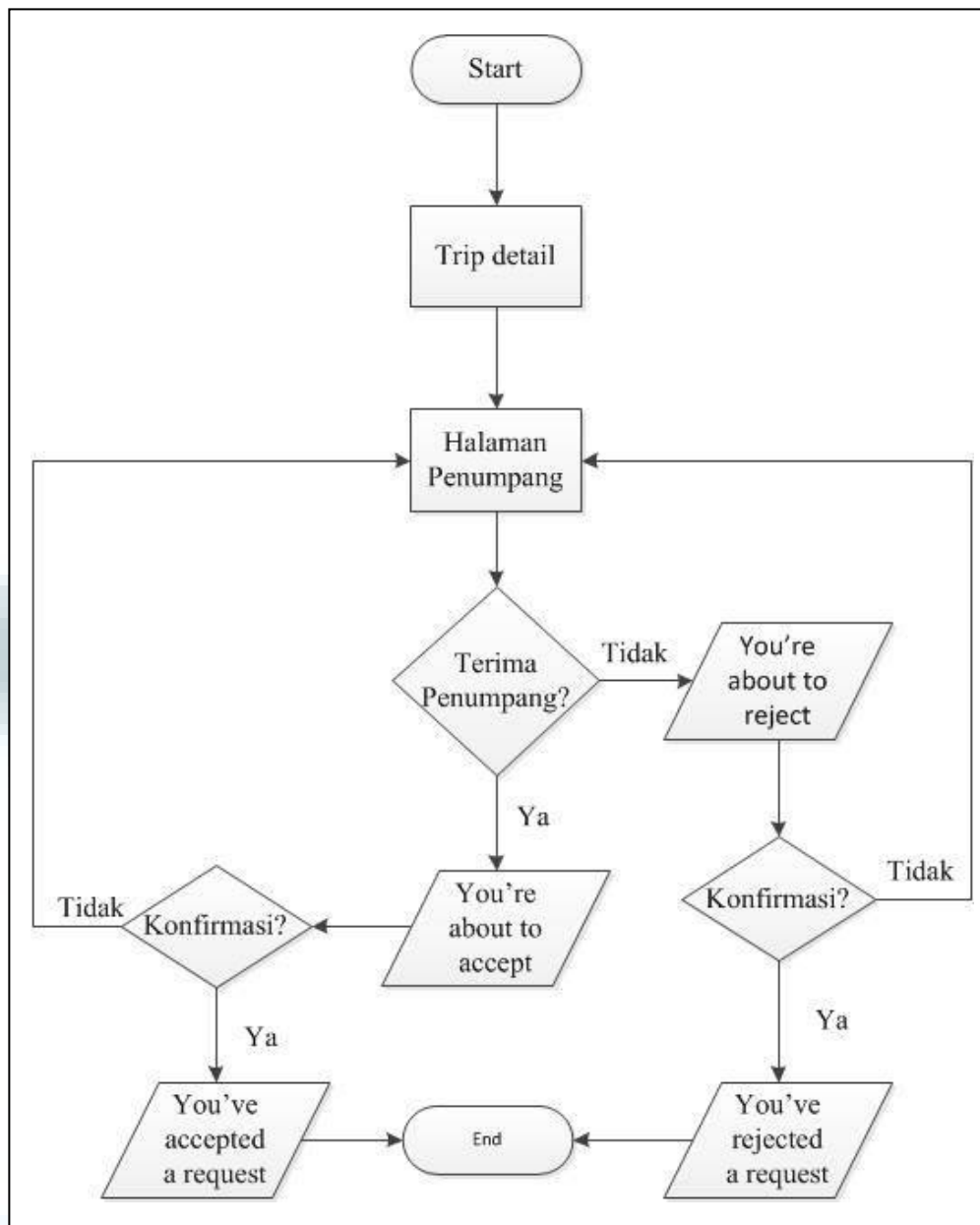


Gambar 3.4 *Flowchart Post Trip*

1.5 Flowchart Accept/Reject Passenger -Driver

Flowchart pada Gambar 3.5 menjelaskan proses *Flowchart Accept/Reject Passenger* dimana proses dimulai dari menekan tombol *trip detail* pada halaman *My Booking*, kemudian *user* dapat melihat detail penumpang yang *request* untuk naik mobil dan *user* diminta menekan tombol *accept/reject*. Jika *user* menekan tombol *accept* maka akan muncul kotak *pop up* “*you’re about to accept*”, Jika *user* menekan tombol *reject* maka akan muncul kotak *pop up* “*you’re about to reject*. Setelah itu *user* diminta konfirmasi untuk *accept/reject*.

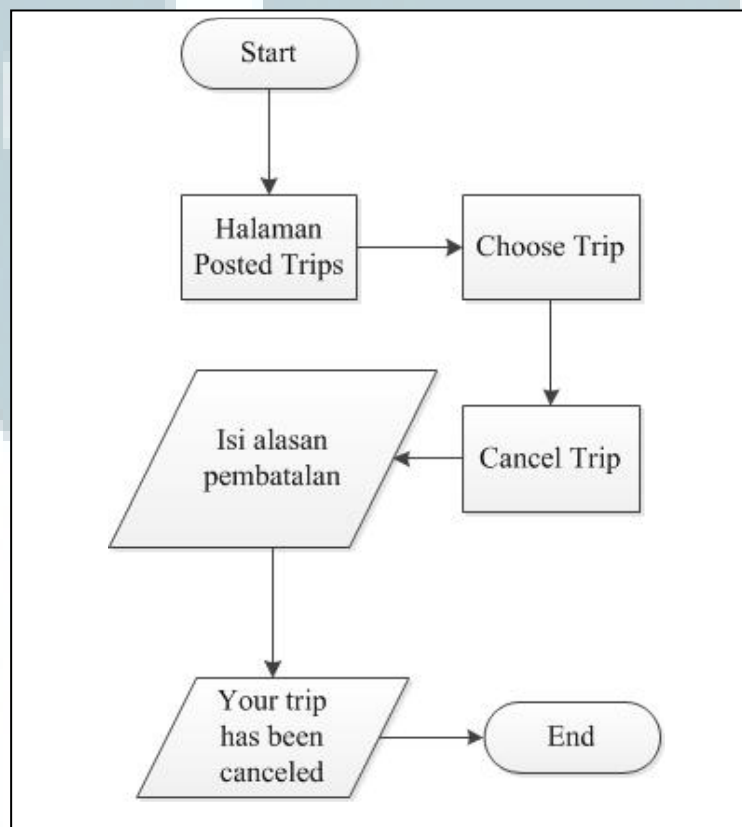
UMN



Gambar 3.5 Flowchart Accept/Reject Passenger

1.6 Flowchart Cancel Trip -Driver

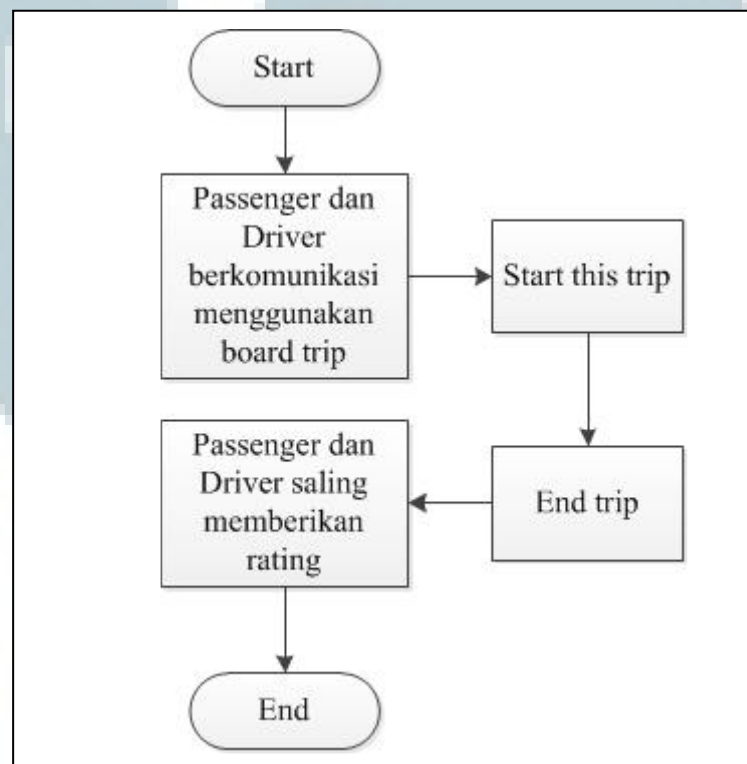
Flowchart pada Gambar 3.6 menjelaskan proses *Flowchart Cancel Trip* dimana proses dimulai dari halaman *Posted Trips* kemudian *user* bisa memilih *trip* yang ingin dibatalkan dengan menekan tombol *cancel trip* setelah menekan tombol *cancel trip* *user* diminta mengisi alasan pembatalan dan konfirmasi atas pembatalan *trip*. Jika *user* sudah konfirmasi atas pembatalan *trip* maka akan muncul kotak *pop up* “*your trip has been canceled*”.



Gambar 3.6 *Flowchart Cancel Trip*

1.7 Flowchart Start and Complete Trip

Flowchart pada Gambar 3.7 menjelaskan proses *Flowchart Start and Complete Trip* dimana proses dimulai dari *Driver* dan *Passenger* saling berkomunikasi melalui *messageboard trip*. Kemudian *Driver* menekan tombol *Start this trip* dan diselesaikan dengan *Driver* menekan tombol *End Trip* pada halaman yang terkunci. Setelah *Trip* selesai maka *Driver* dan *Passenger* harus saling memberikan *rate* jika ingin melakukan *Trip* Berikutnya.

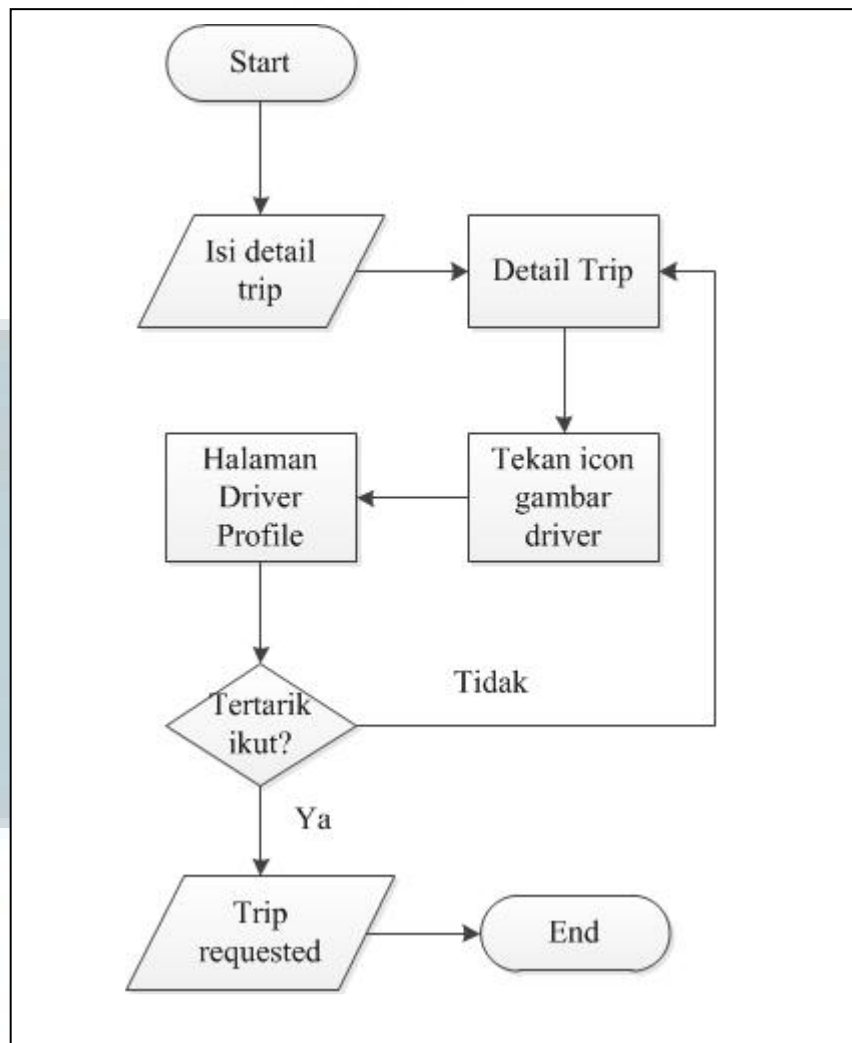


Gambar 3.7 *Flowchart Start and Complete Trip*

1.8 Flowchart Search Trip -Passenger

Flowchart pada Gambar 3.8 menjelaskan proses *Flowchart Search Trip* dimana proses dimulai dari *user* mengisi *field from, meeting radius, to, dropoff radius*, dan *trip detail*. Setelah *user* mengisi semua data yang diperlukan untuk *search trip* maka *user* diminta untuk menekan tombol *Search for trips*. Kemudian akan muncul beberapa pilihan *trip* yang bisa *user* pilih. *User* juga bisa melihat detail tentang *driver* dengan menekan tombol pada gambar *driver*. Pada halaman *driver profile page* akan ada pilihan untuk *join/tidak*. Jika *user* menekan tombol *go back* maka *user* akan kembali halaman *search trip result*, jika *user* check *agree box* maka akan muncul *pop up box "trip requested"*.

UMN

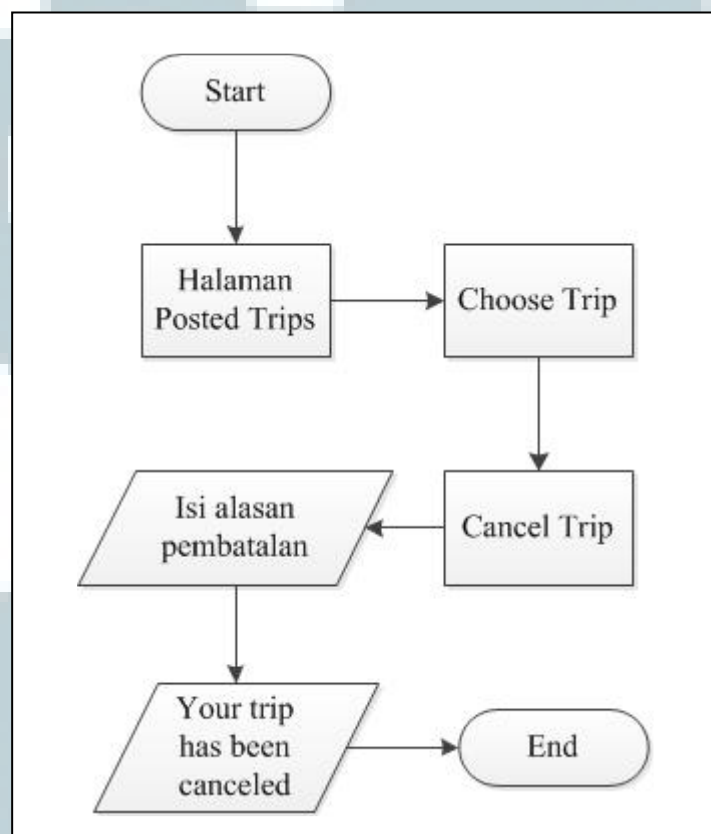


Gambar 3.8 *Flowchart Search Trip*

UMN

1.9 Flowchart Cancel Booking - Passenger

Flowchart pada Gambar 3.9 menjelaskan proses *Flowchart Cancel Booking* dimana proses dimulai dari masuk halaman *Trip Detail* dan *user* menekan tombol *Cancel Booking*. Setelah *user* menekan tombol *Cancel Booking* *user* akan diminta untuk konfirmasi atas pembatalan *Trip*. Jika *user* menekan tombol *Yes* maka pembatalan *trip* berhasil dan dikirim ke halaman *My Booking*. Jika *user* menekan tombol *No* maka *user* akan kembali ke halaman *Trip detail*.



Gambar 3.9 *Flowchart Cancel Booking*

2. Dokumentasi QA(Quality Assurance) Karpul

Quality Assurance dibuat dengan tujuan untuk memberikan hasil *testing* sistem dari aplikasi oleh pihak perusahaan untuk diberikan kepada *client* yang selanjutnya akan dilakukan pengecekan oleh *client* berdasarkan data dari hasil *testing* sistem. Terdapat beberapa *field* dalam dokumentasi *testing* yaitu, Test Cases, Expected Results, Actual Results, Pass/Fail, dan Notes. Test Cases berisi bagian-bagian yang akan di-*test*, Expected Results berisi informasi hasil yang diharapkan dari Test Cases, Actual Results berisi informasi dari hasil Test Cases, Pass/Fail berisi informasi tentang hasil Test Cases berhasil atau gagal, dan Notes berisi pesan dari *tester*. Isi dari bagian QA yang akan di-*testing* dapat dilihat pada lampiran 1.

3. Pengembangan Plugin Manage Stories

Pada proses pengembangan *plugin* diberikan data pendukung dari perusahaan yang berisi data untuk membantu pengembangan *plugin*. Setelah mendapatkan data pendukung akan dilanjutkan ke tahap pengembangan fungsi-fungsi *plugin manage stories* sebagai berikut.

1. *getotherstory*, berfungsi untuk mengambil *field* Story dari tabel *wp_stories* yang diambil berdasarkan perbandingan *Wordid* pada *wp_stories* dengan ID pada *wp_words*.
2. *getword*, berfungsi untuk mengambil *field* Kanji, dan English dari tabel *wp_words* yang diambil berdasarkan perbandingan *Wordid* dengan *Wordid* yang diambil pada halaman utama.
3. *updatestory*, berfungsi untuk mengubah isi dari *field* Story pada tabel *wp_stories* dengan data masukan baru.

4. *insertstory*, berfungsi untuk memasukkan data ke *field* Story pada tabel *wp_stories* dengan data masukan baru.
5. *getstory*, berfungsi untuk mengambil *field* Story dari tabel *wp_stories* yang diambil berdasarkan perbandingan *Userid* pada tabel *wp_stories* dengan ID pada *wp_users*.
6. *copystory*, berfungsi untuk menduplikasi data dari *field* Story pada tabel *wp_stories* milik *user* lain untuk disimpan ke *field* Story pada tabel *wp_stories* milik *user* untuk disimpan.
7. *ratingstory*, berfungsi untuk menyimpan nilai *field* Rating pada tabel *wp_stories* yang didapat dari masukan *user* lain.

Selama proses pengembangan *plugin*, hasil pengembangan dilaporkan kepada pembimbing lapangan untuk dievaluasi dan jika terjadi perubahan dari pembimbing lapangan akan dilakukan revisi terhadap *plugin* tersebut.

Terdapat beberapa tabel yang telah disediakan oleh pembimbing lapangan sehingga dalam proses pembuatannya dikolaborasikan dengan tabel-tabel tersebut dalam proses pengembangan *plugin manage stories*.

3.3.2 Rancangan Sistem

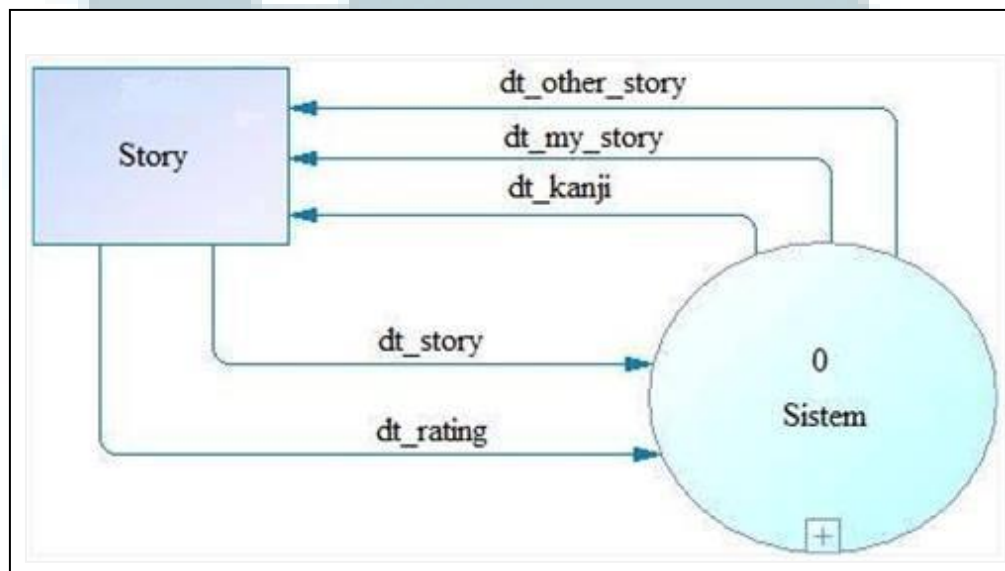
Sebelum melakukan pengembangan *plugin manage stories*, diperlukan analisis dan perancangan terhadap sistem tersebut terlebih dahulu. Tujuannya adalah agar dapat memahami alur kerja sistem tersebut secara keseluruhan serta dapat dilakukan perbaikan fungsi di dalam sistem secara efisien jika terdapat kesalahan.

A. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan diagram yang menjelaskan aliran data yang terdapat dalam suatu sistem secara keseluruhan. Aliran data dalam sistem *plugin manage stories* dapat dijelaskan dalam DFD berikut beserta level-levelnya.

A.1 DFD Level 0

Gambar 3.1 menunjukkan DFD Level 0 pada *plugin manage stories*



Gambar 3.10 DFD Level 0

Sistem *plugin manage stories* memiliki satu entitas yaitu User. Entitas User mempunyai 4 data keluaran yaitu *dt_rating_other_story*, *dt_add_story*, *dt_insert_story*, dan *dt_update_story*, serta mempunyai 3 data masukan yaitu *dt_show_kanji*, *dt_show_other_story*, dan *dt_show_my_story*.



Pada DFD level 1, terdapat 7 proses yang terdiri dari proses Display Kanji, Insert Story, Update Story, Rating Other Story, Show Other Story, Show My Story, dan Copy Story. Proses Display Kanji berguna untuk menampilkan *field* Kanji, English dari tabel wp_words, dan *field* Stories dari tabel wp_stories. Proses Insert Story berguna untuk menambahkan *story* baru pada *field* Stories. Proses Update Story berguna untuk mengubah *field* Stories pada tabel wp_stories dengan data baru. Proses Rating Other Story memberikan nilai pada *field* Rating. Proses Show Other Story berguna untuk menampilkan data pada *field* Story milik *user* lain. Proses Show My Story berguna untuk menampilkan data pada *field* Story milik *user*. Proses Copy Story mengambil data *story* milik *user* lain untuk dimasukkan ke *field* Story milik *user* pada tabel wp_stories.

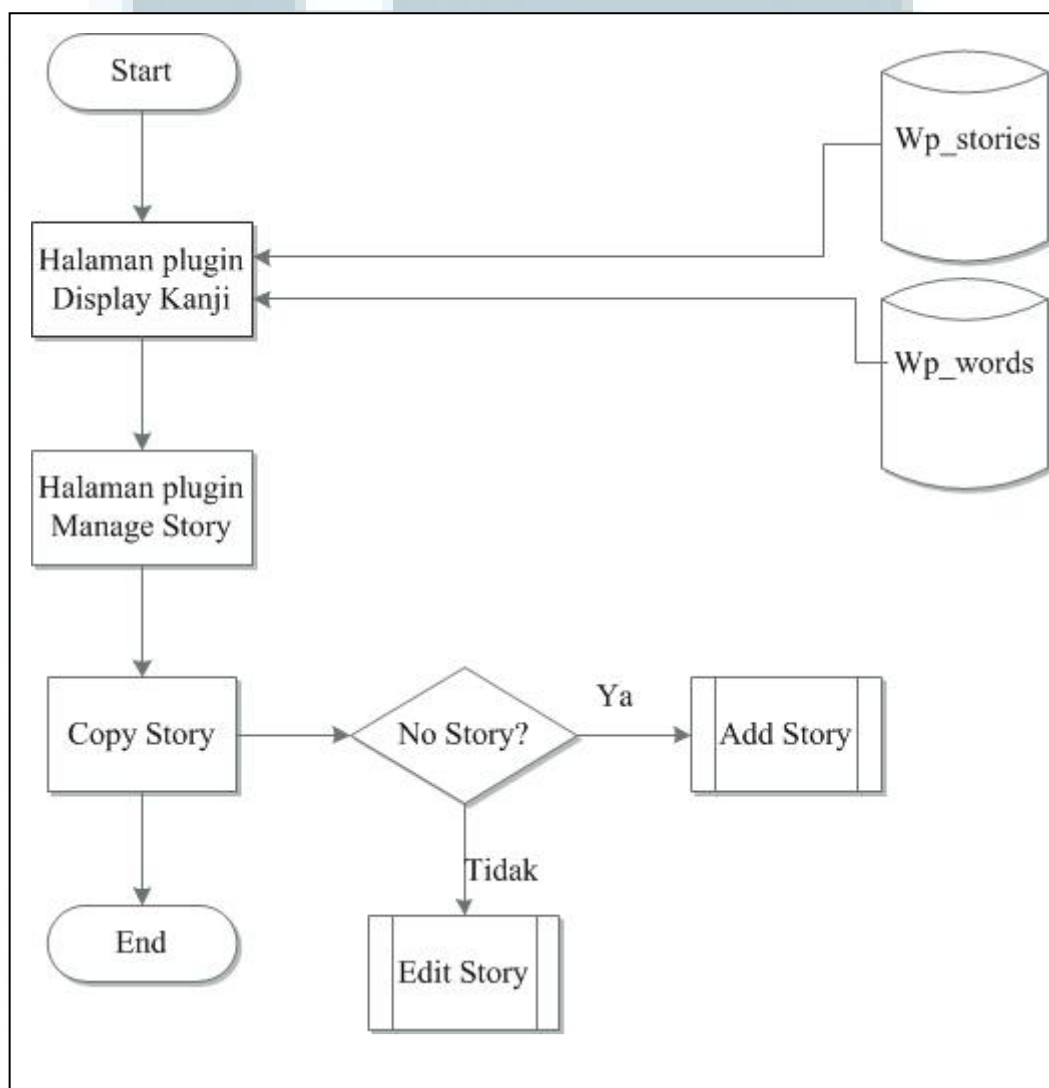
C. Flowchart Diagram

Flowchart diagram menjelaskan keseluruhan alur kerja dari suatu sistem dari awal sampai akhir. Alur kerja sistem dapat dijelaskan dalam *flowchart* berikut ini.

UMN

C.1 Flowchart Copy Story

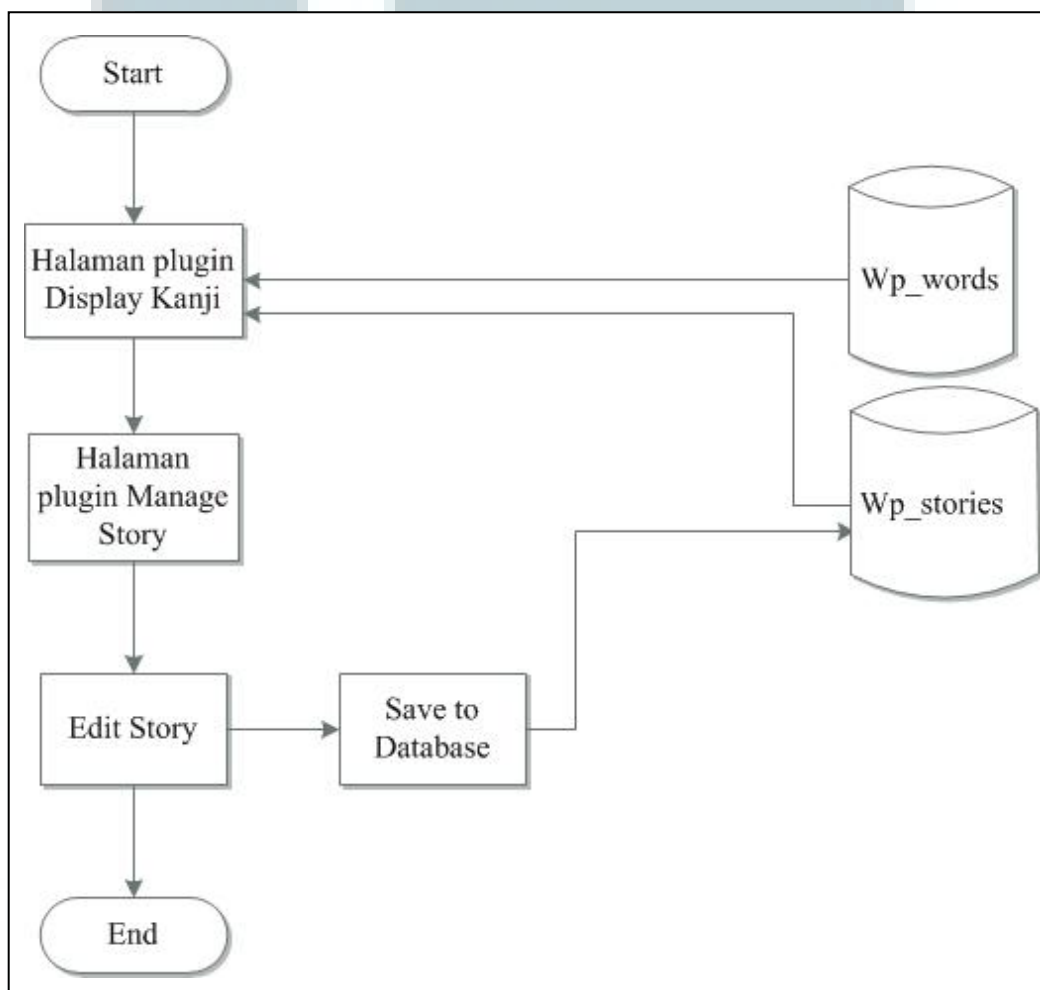
Flowchart pada Gambar 3.12 menjelaskan proses Copy Story dimana program akan dimulai dari halaman utama dan jika *user* menekan salah satu data pada tabel maka akan menampilkan halaman *story* yang akan menampilkan *story* milik *user* dan *story user* lainnya tentang kata tersebut. Kemudian bisa dilakukan *Copy Story* milik *user* lain dengan mengklik tombol *Copy Story*, data yang telah di-copy akan disimpan ke tabel *wp_stories* pada *database* *project_mirai*.



Gambar 3.12 *Flowchart Copy Story*

C.2 Flowchart Edit Story

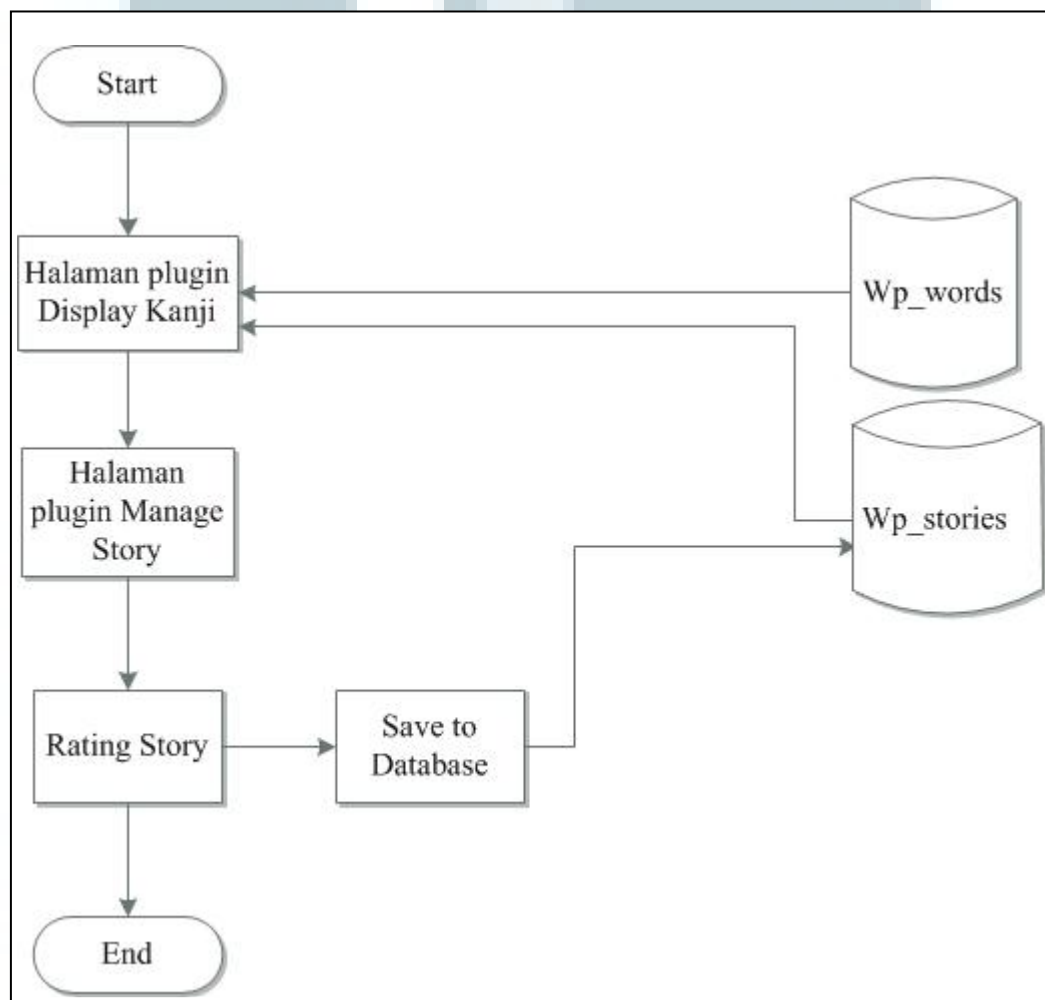
Flowchart pada Gambar 3.13 menjelaskan proses Edit Story dimana program akan dimulai dari halaman utama dan jika *user* menekan salah satu data pada tabel maka akan menampilkan halaman *story* yang akan menampilkan *story* milik *user* dan *story user* lainnya tentang kata tersebut. Kemudian bisa dilakukan *edit story* milik *user* dengan mengubah isi dari *story* kemudian menekan tombol *Save*, lalu data akan disimpan ke tabel *wp_stories* pada *database* *project_mirai*.



Gambar 3.13 *Flowchart Edit Story*

C.3 Flowchart Rating Story

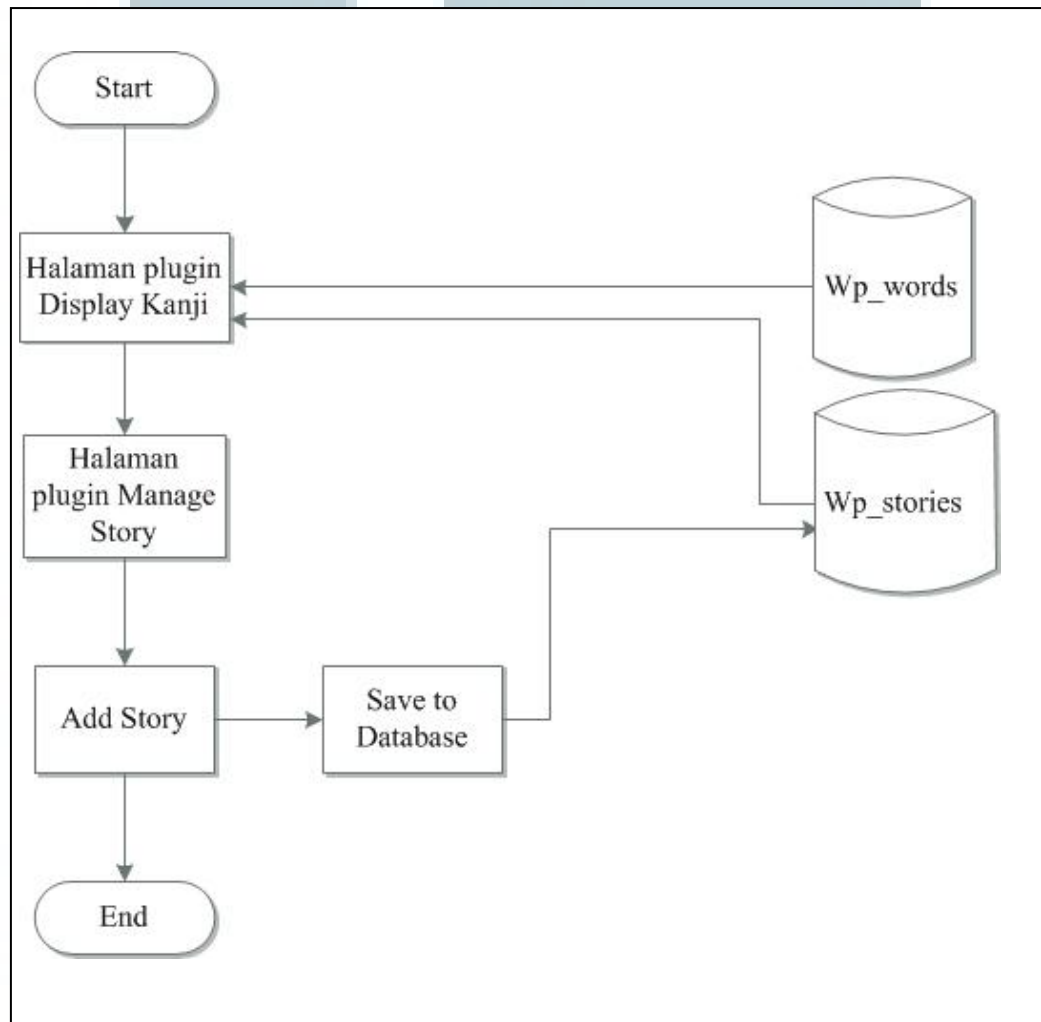
Flowchart pada Gambar 3.14 menjelaskan proses Rating Story dimana program akan dimulai dari halaman utama dan jika *user* menekan salah satu data pada tabel maka akan menampilkan halaman *story* yang akan menampilkan *story* milik *user* dan *story user* lainnya tentang kata tersebut. Kemudian bisa dilakukan *Rating* milik *user* lain dengan mengklik pada *icon* bintang yang memiliki nilai 1-5, setelah *icon* diklik maka hasil rating akan disimpan ke tabel *wp_stories* pada *database* *project_mirai*.



Gambar 3.14 *Flowchart Rating Story*

C.4 Flowchart Add Story

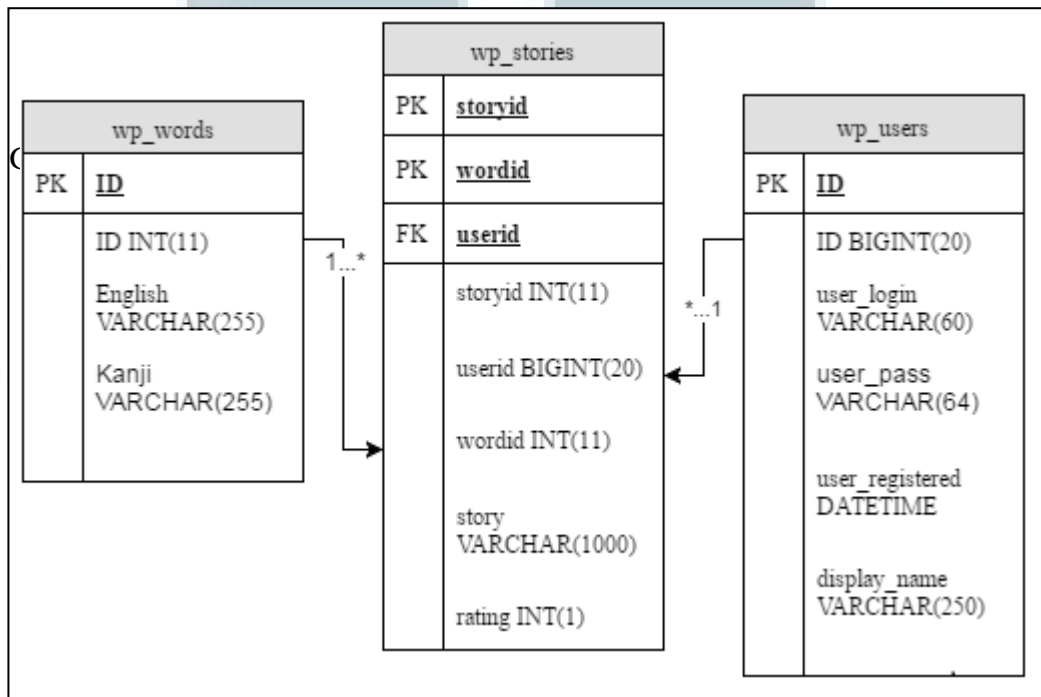
Flowchart pada Gambar 3.15 menjelaskan proses Add Story dimana program akan dimulai dari halaman utama dan jika *user* menekan salah satu data pada tabel maka akan menampilkan halaman *story* yang akan menampilkan *story* milik *user* dan *story user* lainnya tentang kata tersebut. Kemudian bisa dilakukan *Add Story* milik *user* dengan mengisi isi dari *story* kemudian menekan tombol *Save*, lalu data akan disimpan ke tabel *wp_stories* pada *database* *project_mirai*.



Gambar 3.15 *FlowchartAdd Story*

D. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) menjelaskan tentang hubungan antara tabel-tabel dalam *database* yang digunakan selama proses pengembangan *plugin manage stories* sistem Hello Huayu. Gambar 3.16 menunjukkan ERD *plugin manage stories*.



Gambar 3.16 ERD *plugin manage stories*

E. Struktur Tabel

Dalam proses pengembangan *plugin manage stories* pada sistem Hello Huayu terdapat 2 tabel yang digunakan yaitu tabel **wp_words** dan **wp_stories**. **wp_words** menyimpan data-data kanji dan **wp_stories** menyimpan story-story yang dibuat oleh *user*.

Nama : wp_words

Fungsi : menyimpan data-data kanji

Tabel 3.2 Struktur Tabel wp_words

Field Name	Type	Length	Information
ID	Int	11	Primary Key, id word
Kanji	Varchar	255	Data Kanji
English	Varchar	255	Translate Inggris

Nama : wp_stories

Fungsi : menyimpan *story user*

Tabel 3.3 Struktur Tabel wp_stories

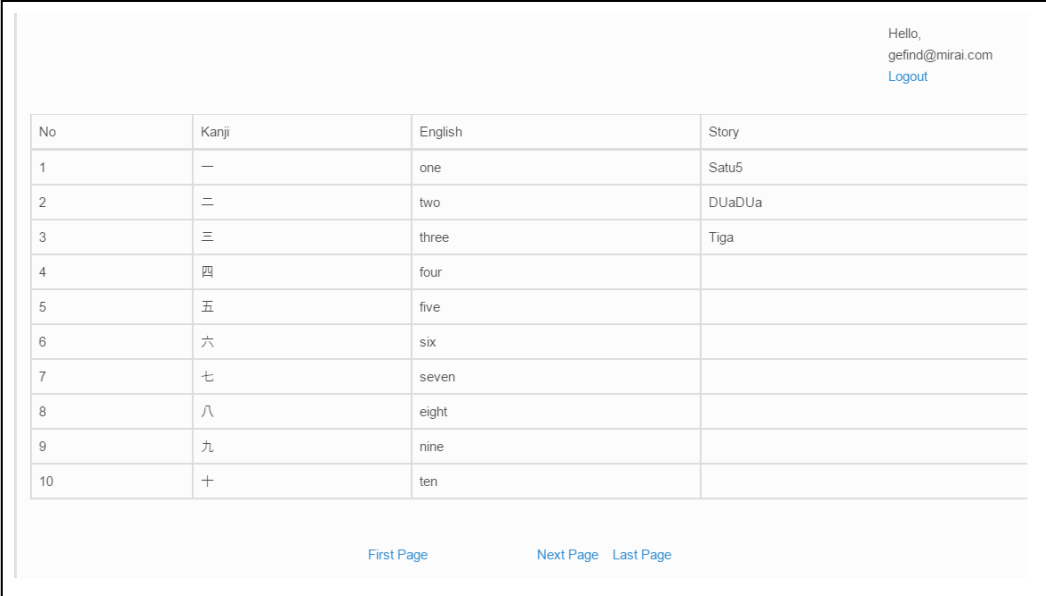
Field Name	Type	Length	Information
Storyid	Int	11	Primary Key, id story
Userid	Bigint	20	Foreign key, id user
Wordid	Int	11	Foreign key, id words
Stories	Varchar	1000	Story User
Rating	Int	1	Rating

3.3.3 Implementasi

Berdasarkan rancangan *flowchart* yang telah dibuat. Pada tahap ini dilakukan juga proses analisis terkait hal-hal yang diperlukan selama proses pengembangan *plugin*. Selain itu dilakukan pembuatan program dengan menggunakan PHP, HTML, CSS, Javascript dan JQuery. Pembuatan program diawali dengan menyambungkan *plugin* tersebut ke dalam *database* MYSQL dan kemudian dilakukan proses pengambilan data dari *database* ke dalam *plugin* atau dilakukan proses perubahan data dari *plugin* ke dalam *database* project_mirai.

A. Halaman Plugin Display Kanji

Gambar 3.17 menunjukkan halaman utama dari sistem Hello Huayu yang di dalamnya terdapat menu *Logout*, *First Page*, *Next Page*, *Last Page*. Terdapat daftar data kata kanji yang memiliki *field* berisi No, Kanji, *English*, dan *Story*. Menu *Logout* berfungsi untuk keluar dari sistem, *First Page* berfungsi untuk menampilkan halaman pertama dari tabel *wp_stories*, *Next Page* berfungsi untuk menampilkan halaman lanjutan dari tabel *wp_stories*, dan *Last Page* berfungsi untuk menampilkan halaman terakhir dari tabel *wp_stories*. Jika data di dalam tabel diklik maka akan menampilkan halaman yang dapat dilihat pada Gambar 3.17.



The screenshot shows a web interface for the 'Hello Huayu' plugin. In the top right corner, there is a user greeting 'Hello, gefind@mirai.com' and a blue 'Logout' link. Below this is a table with four columns: 'No', 'Kanji', 'English', and 'Story'. The table contains 10 rows of data, numbered 1 to 10, listing the first ten kanji characters (一 through 十) with their English equivalents and a 'Story' field. At the bottom of the table, there are three blue navigation links: 'First Page', 'Next Page', and 'Last Page'.

No	Kanji	English	Story
1	一	one	Satu5
2	二	two	DUaDUa
3	三	three	Tiga
4	四	four	
5	五	five	
6	六	six	
7	七	seven	
8	八	eight	
9	九	nine	
10	十	ten	

First Page Next Page Last Page

Gambar 3.17 Halaman *Plugin Display Kanji*

B. Halaman Plugin Manage Stories

Gambar 3.18 menunjukkan *story* dari data Kanji. Terdapat 2 tabel pada halaman ini yaitu tabel untuk menampilkan detail *story* diri sendiri yang terletak di bagian atas, dan tabel untuk menampilkan *story* orang lain tentang kanji yang sama yang terletak pada bagian bawah. Pada tabel yang di atas memiliki tombol Save yang berfungsi untuk menyimpan *update story*. Pada tabel yang di bawah memiliki tombol Copy Story yang berfungsi untuk meng-*copystory* yang diminati, dan juga memiliki fitur *Rating* yang berfungsi untuk melakukan *rate* terhadap *story* tersebut.

Hello Huayu

Kanji	English	Story
一	one	Satu1

Save

Name	Story		
george	Satu2	Copy Story	Rating: ★★★★★
david	Satu3	Copy Story	Rating: ★★★★★☆
gefind	Satu4	Copy Story	Rating: ★★★★★☆
hary	Satu5	Copy Story	Rating: ★★★★★☆
jason	Satu6	Copy Story	Rating: ★★★★★☆
tyan	Satu7	Copy Story	Rating: ★★★★★☆

Gambar 3.18 Halaman *Plugin Manage Stories*

3.4 Kendala yang Ditemukan

Kendala-kendala yang ditemukan selama proses kerja magang di PT Mirai Internasional Indonesia adalah sebagai berikut.

1. Belum terdapat standar pembuatan QA (*Quality Assurance*) pada PT Mirai Internasional Indonesia sehingga menyulitkan dalam proses pembuatan QA.
2. Dalam pembuatan *flowchart* aplikasi Karpul *client* belum mendapatkan gambaran pasti tentang aplikasi yang ingin dibuat.
3. Terjadi perubahan dalam pengembangan *plugin* sistem Hello Huayu yang sebelumnya menggunakan bahasa pemrograman ROR (Ruby On Rails) menjadi PHP.

3.4 Solusi atas Kendala yang Ditemukan

Solusi untuk kendala-kendala yang ditemukan dalam proses kerja magang di PT Mirai Internasional Indonesia adalah sebagai berikut.

1. Dilakukan pertemuan dengan pembimbing lapangan untuk membahas QA yang sesuai dengan keinginan standar perusahaan.
2. Memberikan beberapa pilihan *flowchart* sehingga memudahkan *client* memutuskan *flowchart* yang paling sesuai.
3. Mengubah bahasa pemrograman dari ROR menjadi PHP.